|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **STUDENT SIGN** |
| **Md. Sazzad Khan** | **22-49172-3** |  |

**Instructions:**

* **Make sure to write your Name, ID and Signature on this document.**
* **First write your signature on a paper then take photo of that signature and use it for signing this document.**
* **After completing the requirements of the midterm assignment by editing this document, upload this document in the link provided in your VUES Student Account.**
* **Submission Deadline: 18th August 2025, 11:59pm.**

**Midterm Assignment**

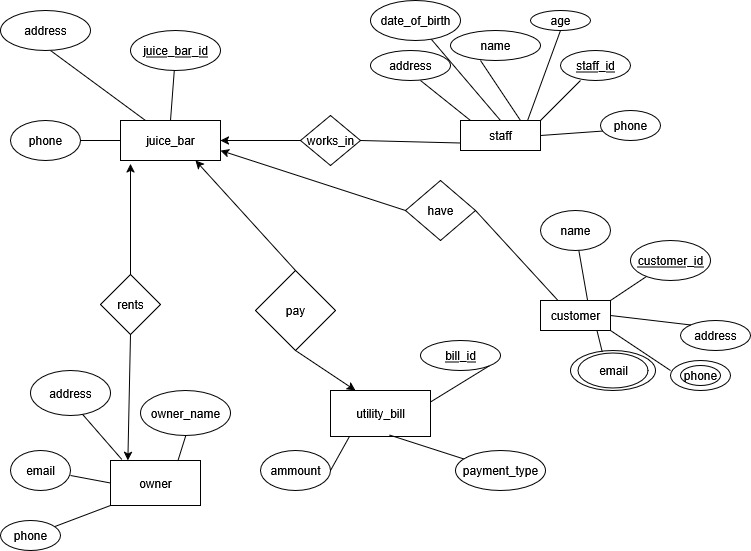
1. **Below a scenario has been given draw the ER Diagram.**

***Draw with proper annotations (use DIA, VISIO, MS WORD etc.).***

***For reference see ERDiagramTutorial.***

In a Juice Bar Management System, one Juice Bar may have many staff. But one staff can work in one Juice Bar only. Each staff has a unique identification number, name, age, address, date of birth and phone number. Juice Bar has a unique identification number, address, and phone. One Juice Bar maybe rented by exactly one owner. One owner may rent exactly one Juice Bar. Owner is defined by name, address, email and phone. A Juice Bar may have many customers. Each customer has a unique identification number, name, address, email, phone. Each customer can have more than one email address and phone number. A Juice Bar must pay the utility bill which has a unique identification number, payment\_type and amount.

Answer Box 1:



1. **Below an ER Diagram has been given write the scenario.**

***For reference see ERDiagramTutorial.***

A diagram of a company

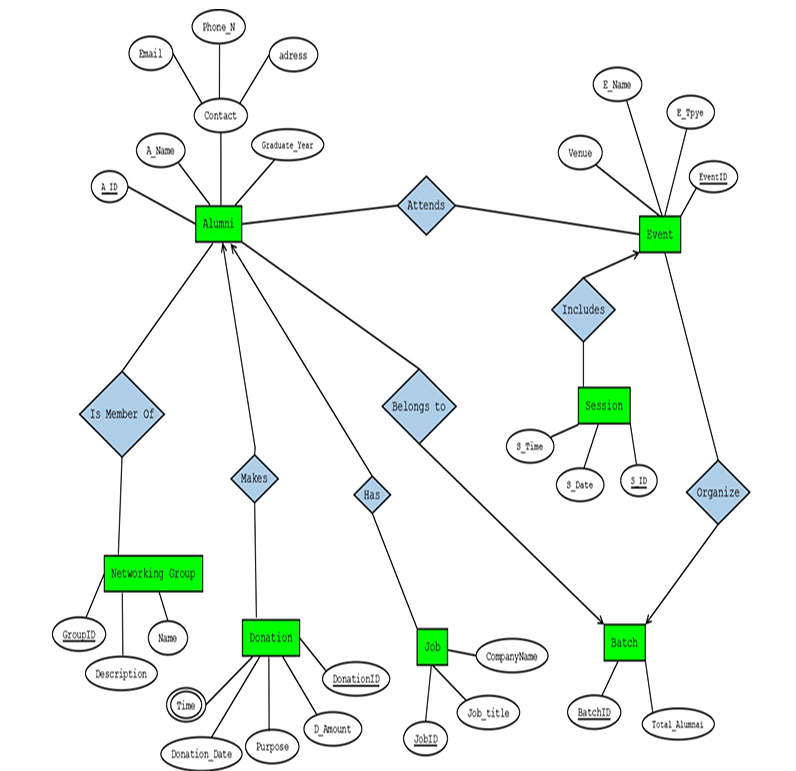
Description automatically generated

Answer Box 2:

|  |
| --- |
| A Bus Transportation Ticketing System that coordinates the activities of buses, routes, staff, and passengers. In this system, each bus is uniquely identified by a Bus Number and has specific details such as its type, model, and capacity. Every bus follows a particular route, which is identified by a Route ID and includes the starting location,ending location, distance, and departure time.  The operation of buses involves drivers and supervisors. Bus is operate by driver.Each driver is uniquely identified by a Driver ID, and has a first name, last name, license number, and years of experience. Supervisors, identified by a Supervisor ID, are responsible for overseeing operations.Supervisors conduct the bus operation and issuing tickets. Each supervisor has a first name, last name, and experience in years.  Passengers in the system are identified using a National ID and have attributes such as name, gender, and more than one phone number.Passengers purchase ticket.every ticket has ticket statuse,price,seat number anda unique ticket number. When a passenger purchases a ticket, a transaction is recorded. A transaction includes a unique Transaction ID, the date of the transaction, and the total amount paid. Each transaction is associated with a particular payment type, defined by a unique Payment Code, Payment Type, and Payment Method.  Tickets are central to the system and are uniquely identified by a Ticket Number. Each ticket includes a seat number, status and price. Tickets are issued to passengers, linked to specific buses, and managed by supervisors. This structure allows the system to efficiently track which passengers are traveling on which buses, using which routes, and how payments are made and recorded. |

1. **Normalize the ER Diagram given below up to 3rd Normal Form and finalize the tables that needs to be created. Then (in Oracle using SQL) write down the queries that are required to create all the tables with necessary constraints. Also insert at least 3 rows of data in each created table.**

***For reference see NormalizationTutorial and BasicSQLTutorial.***



Answer Box 3

|  |
| --- |
| **Normalization**  **Relation: Belong to**  **UNF**  Belong to(A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Batch\_ID, Total\_Alumni)  **1NF**  There is no multi-valued attribute. Relation already in 1NF.   1. A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Batch\_ID, Total\_Alumni.   **2NF**   1. A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address. 2. Batch\_ID, Total\_Alumni.   **3NF**   1. A\_ID, A\_Name, Graduate\_Year. 2. Email, Phone\_N, Address. 3. Batch\_ID, Total\_Alumni.   **Table Creation**   1. A\_ID, A\_Name, Graduate\_Year, **Email**. 2. Email, Phone\_N, Address. 3. Batch\_ID, Total\_Alumni. 4. **A\_ID**, **Batch\_ID.**   **Relation: Attends**  **UNF**  Attends(A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Event\_ID, E\_Name, E\_Type, Venue)  **1NF**  There is no multi-valued attribute. Relation already in 1NF.   1. A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Event\_ID , E\_Name, E\_Type, Venue.   **2NF**   1. A\_ID,A\_Name,Graduate\_Year,Email,Phone\_N,Address. 2. Event\_ID , E\_Name, E\_Type, Venue.   **3NF**   1. A\_ID, A\_Name, Graduate\_Year. 2. Email, Phone\_N, Address. 3. Event\_ID, E\_Name, E\_Type, Venue.   **Table Creation**   1. A\_ID, A\_Name, Graduate\_Year, **Email**. 2. Email, Phone\_N, Address. 3. Event\_ID, E\_Name, E\_Type, Venue. 4. **A\_ID, Event\_ID.**   **Relation: Is Member Of**  **UNF**  Is Member Of (A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Group\_ID, G\_Name, Description)  **1NF**  There is no multi-valued attribute. Relation already in 1NF.   1. A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Group\_ID, G\_Name, Description.   **2NF**   1. A\_ID,A\_Name,Graduate\_Year,Email,Phone\_N,Address. 2. Group\_ID, G\_Name, Description.   **3NF**   1. A\_ID, A\_Name, Graduate\_Year. 2. Email, Phone\_N, Address. 3. Group\_ID, G\_Name, Description.   **Table Creation**   1. A\_ID, A\_Name, Graduate\_Year, **Email**. 2. Email, Phone\_N, Address. 3. Group\_ID, G\_Name, Description. 4. **A\_ID, Group\_ID.**   **Relation: Makes**  **UNF**  Makes (A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Donation\_ID, D\_Amount, Donation\_Date, Purpose, Time)  **1NF**  Time is a multi valued attribute.   1. A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Donation\_ID, D\_Amount, Donation\_Date, Purpose, Time.   **2NF**   1. A\_ID,A\_Name,Graduate\_Year,Email,Phone\_N,Address. 2. Donation\_ID, D\_Amount, Donation\_Date, Purpose, Time.   **3NF**   1. A\_ID, A\_Name, Graduate\_Year. 2. Email, Phone\_N, Address. 3. Donation\_ID, D\_Amount, Donation\_Date, Purpose, Time.   **Table Creation**   1. A\_ID, A\_Name, Graduate\_Year, **Email**. 2. Email, Phone\_N, Address. 3. Donation\_ID, D\_Amount, Donation\_Date, Purpose. 4. **Donation\_ID**, Time. 5. **A\_ID, Donation\_ID.**   **Relation: Has**  **UNF**  Has (A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Job\_ID, Job\_title, CompanyName)  **1NF**  There is no multi-valued attribute. Relation already in 1NF.   1. A\_ID, A\_Name, Graduate\_Year, Email, Phone\_N, Address, Job\_ID, Job\_title, CompanyName**.**   **2NF**   1. A\_ID,A\_Name,Graduate\_Year,Email,Phone\_N,Address. 2. Job\_ID, Job\_title, CompanyName.   **3NF**   1. A\_ID, A\_Name, Graduate\_Year. 2. Email, Phone\_N, Address. 3. Job\_ID, Job\_title, CompanyName.   **Table Creation**   1. A\_ID, A\_Name, Graduate\_Year, **Email**. 2. Email, Phone\_N, Address. 3. Job\_ID, Job\_title, CompanyName. 4. **A\_ID, Job\_ID.**   **Relation: Includes**  **UNF**  Includes (Event\_ID, E\_Name, E\_Type, Venue, S\_ID, S\_Time, S\_Date)  **1NF**  There is no multi-valued attribute. Relation already in 1NF.   1. Event\_ID, E\_Name, E\_Type, Venue, S\_ID, S\_Time, S\_Date.   **2NF**   1. Event\_ID, E\_Name, E\_Type, Venue. 2. S\_ID, S\_Time, S\_Date   **3NF**   1. Event\_ID, E\_Name, E\_Type, Venue. 2. S\_ID, S\_Time, S\_Date.   **Table Creation**   1. Event\_ID, E\_Name, E\_Type, Venue. 2. S\_ID, S\_Time, S\_Date 3. **Event\_ID**, **S\_ID**.   **Relation: Organize**  **UNF**  Organize (Event\_ID, E\_Name, E\_Type, Venue, Batch\_ID, Total\_Alumni)  **1NF**  There is no multi-valued attribute. Relation already in 1NF.   1. Event\_ID, E\_Name, E\_Type, Venue, Batch\_ID, Total\_Alumni.   **2NF**   1. Event\_ID, E\_Name, E\_Type, Venue. 2. Batch\_ID, Total\_Alumni.   **3NF**   1. Event\_ID, E\_Name, E\_Type, Venue. 2. Batch\_ID, Total\_Alumni.   **Table Creation**   1. Event\_ID, E\_Name, E\_Type, Venue. 2. Batch\_ID, Total\_Alumni**.** 3. **Event\_ID**, **Batch\_ID**.   **Final Table Creation:**   1. **Alumni:** A\_ID, A\_Name, Graduate\_Year, **Email**. 2. **Contact:** Email, Phone\_N, Address. 3. **Batch:** Batch\_ID, Total\_Alumni. 4. **Alumni\_Batch: A\_ID**, **Batch\_ID.** 5. **Event:** Event\_ID, E\_Name, E\_Type, Venue. 6. **Alumni\_Event: A\_ID, Event\_ID.** 7. **Networking\_Group:** Group\_ID, G\_Name, Description. 8. **Alumni\_Group: A\_ID, Group\_ID.** 9. **Donation:** Donation\_ID, D\_Amount, Donation\_Date, Purpose. 10. **Donation\_Time: Donation\_ID**, Time. 11. **Alumni\_Donation: A\_ID, Donation\_ID.** 12. **Job:** Job\_ID, Job\_title, CompanyName. 13. **Alumni\_Job: A\_ID, Job\_ID.** 14. **Session\_Date:** S\_ID, S\_Time, S\_Date. 15. **Event\_Session: Event\_ID**, **S\_ID**. 16. **Event\_Batch: Event\_ID**, **Batch\_ID**.   **Final Table Creation Query**   1. **Alumni:**   CREATE TABLE Alumni (  A\_ID NUMBER PRIMARY KEY,  A\_Name VARCHAR2(100) NOT NULL,  Graduate\_Year NUMBER(4) NOT NULL,  Email VARCHAR2(100) UNIQUE NOT NULL  );   1. **Contact:**   CREATE TABLE Contact (  Email VARCHAR2(100) PRIMARY KEY,  Phone\_N VARCHAR2(20) NOT NULL,  Address VARCHAR2(200)  );  ALTER TABLE Alumni  ADD CONSTRAINT fk\_alumni\_contact FOREIGN KEY (Email)  REFERENCES Contact(Email) ON DELETE CASCADE;   1. **Batch:**   CREATE TABLE Batch (  Batch\_ID NUMBER PRIMARY KEY,  Total\_Alumni NUMBER NOT NULL  );   1. **Alumni\_Batch:**   CREATE TABLE Alumni\_Batch (  A\_ID NUMBER,  Batch\_ID NUMBER,  PRIMARY KEY (A\_ID, Batch\_ID),  FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID) ON DELETE CASCADE,  FOREIGN KEY (Batch\_ID) REFERENCES Batch(Batch\_ID) ON DELETE CASCADE  );   1. **Event:**   CREATE TABLE Event (  Event\_ID NUMBER PRIMARY KEY,  E\_Name VARCHAR2(100) NOT NULL,  E\_Type VARCHAR2(50) NOT NULL,  Venue VARCHAR2(100) NOT NULL  );   1. **Alumni\_Event:**   CREATE TABLE Alumni\_Event (  A\_ID NUMBER,  Event\_ID NUMBER,  PRIMARY KEY (A\_ID, Event\_ID),  FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID) ON DELETE CASCADE,  FOREIGN KEY (Event\_ID) REFERENCES Event(Event\_ID) ON DELETE CASCADE  );   1. **Networking\_Group:**   CREATE TABLE Networking\_Group (  Group\_ID NUMBER PRIMARY KEY,  G\_Name VARCHAR2(100) NOT NULL,  Description VARCHAR2(200) NULL  );   1. **Alumni\_Group:**   CREATE TABLE Alumni\_Group (  A\_ID NUMBER,  Group\_ID NUMBER,  PRIMARY KEY (A\_ID, Group\_ID),  FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID) ON DELETE CASCADE,  FOREIGN KEY (Group\_ID) REFERENCES Networking\_Group(Group\_ID) ON DELETE CASCADE  );   1. **Donation:**   CREATE TABLE Donation (  Donation\_ID NUMBER PRIMARY KEY,  D\_Amount NUMBER(10,2) NOT NULL,  Donation\_Date DATE NOT NULL,  Purpose VARCHAR2(200) NOT NULL  );   1. **Donation\_Time:**   CREATE TABLE Donation\_Time (  Donation\_ID NUMBER,  Time VARCHAR2(20) NOT NULL,  PRIMARY KEY (Donation\_ID, Time),  FOREIGN KEY (Donation\_ID) REFERENCES Donation(Donation\_ID) ON DELETE CASCADE  );   1. **Alumni\_Donation:**   CREATE TABLE Alumni\_Donation (  A\_ID NUMBER,  Donation\_ID NUMBER,  PRIMARY KEY (A\_ID, Donation\_ID),  FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID) ON DELETE CASCADE,  FOREIGN KEY (Donation\_ID) REFERENCES Donation(Donation\_ID) ON DELETE CASCADE  );   1. **Job:**   CREATE TABLE Job (  Job\_ID NUMBER PRIMARY KEY,  Job\_Title VARCHAR2(100) NOT NULL,  CompanyName VARCHAR2(100) NOT NULL  );   1. **Alumni\_Job:**   CREATE TABLE Alumni\_Job (  A\_ID NUMBER,  Job\_ID NUMBER,  PRIMARY KEY (A\_ID, Job\_ID),  FOREIGN KEY (A\_ID) REFERENCES Alumni(A\_ID) ON DELETE CASCADE,  FOREIGN KEY (Job\_ID) REFERENCES Job(Job\_ID) ON DELETE CASCADE  );   1. **Session\_Date:**   CREATE TABLE Session\_Date (  S\_ID NUMBER PRIMARY KEY,  S\_Time VARCHAR2(20) NOT NULL,  S\_Date DATE NOT NULL  );   1. **Event\_Session:**   CREATE TABLE Event\_Session (  Event\_ID NUMBER,  S\_ID NUMBER,  PRIMARY KEY (Event\_ID, S\_ID),  FOREIGN KEY (Event\_ID) REFERENCES Event(Event\_ID) ON DELETE CASCADE,  FOREIGN KEY (S\_ID) REFERENCES Session\_Date(S\_ID) ON DELETE CASCADE  );   1. **Event\_Batch:**   CREATE TABLE Event\_Batch (  Event\_ID NUMBER,  Batch\_ID NUMBER,  PRIMARY KEY (Event\_ID, Batch\_ID),  FOREIGN KEY (Event\_ID) REFERENCES Event(Event\_ID) ON DELETE CASCADE,  FOREIGN KEY (Batch\_ID) REFERENCES Batch(Batch\_ID) ON DELETE CASCADE  );  **Data Insertion Query**   1. INSERT INTO Contact VALUES ('sazzad@aiub.edu', '01710000001', 'Dhaka, Bangladesh'); 2. INSERT INTO Contact VALUES ('shahriar@aiub.edu', '01710000002', 'Chittagong, Bangladesh'); 3. INSERT INTO Contact VALUES ('ismail@aiub.edu', '01710000003', 'Sylhet, Bangladesh'); 4. INSERT INTO Alumni VALUES (1, 'Md Sazzad Khan', 2022, 'sazzad@aiub.edu'); 5. INSERT INTO Alumni VALUES (2, 'Hashamee Al Shahriar', 2021, 'shahriar@aiub.edu'); 6. INSERT INTO Alumni VALUES (3, 'Ismail Hossain', 2020, 'ismail@aiub.edu'); 7. INSERT INTO Batch VALUES (101, 50); 8. INSERT INTO Batch VALUES (102, 60); 9. INSERT INTO Batch VALUES (103, 70); 10. INSERT INTO Alumni\_Batch VALUES (1, 101); 11. INSERT INTO Alumni\_Batch VALUES (2, 102); 12. INSERT INTO Alumni\_Batch VALUES (3, 103); 13. INSERT INTO Event VALUES (201, 'Alumni Meet 2023', 'Conference', 'AIUB Campus Hall'); 14. INSERT INTO Event VALUES (202, 'Tech Fest', 'Seminar', 'AIUB Auditorium'); 15. INSERT INTO Event VALUES (203, 'Job Fair', 'Career', 'BICC Dhaka'); 16. INSERT INTO Alumni\_Event VALUES (1, 201); 17. INSERT INTO Alumni\_Event VALUES (2, 202); 18. INSERT INTO Alumni\_Event VALUES (3, 203); 19. INSERT INTO Networking\_Group VALUES (301, 'AIUB Techies', 'Technology-focused alumni group'); 20. INSERT INTO Networking\_Group VALUES (302, 'AIUB Entrepreneurs', 'Startup and business discussions'); 21. INSERT INTO Networking\_Group VALUES (303, 'AIUB Researchers', NULL); 22. INSERT INTO Alumni\_Group VALUES (1, 301); 23. INSERT INTO Alumni\_Group VALUES (2, 302); 24. INSERT INTO Alumni\_Group VALUES (3, 303); 25. INSERT INTO Donation VALUES (401, 10000, TO\_DATE('2024-05-20','YYYY-MM-DD'), 'Library Fund'); 26. INSERT INTO Donation VALUES (402, 5000, TO\_DATE('2024-06-15','YYYY-MM-DD'), 'Scholarship'); 27. INSERT INTO Donation VALUES (403, 7000, TO\_DATE('2024-07-10','YYYY-MM-DD'), 'Sports Development'); 28. INSERT INTO Donation\_Time VALUES (401, '10:30 AM'); 29. INSERT INTO Donation\_Time VALUES (401, '11:00 AM'); 30. INSERT INTO Donation\_Time VALUES (401, '01:15 PM'); 31. INSERT INTO Donation\_Time VALUES (402, '02:00 PM'); 32. INSERT INTO Donation\_Time VALUES (402, '03:45 PM'); 33. INSERT INTO Donation\_Time VALUES (402, '04:10 PM'); 34. INSERT INTO Donation\_Time VALUES (403, '05:15 PM'); 35. INSERT INTO Donation\_Time VALUES (403, '06:30 PM'); 36. INSERT INTO Donation\_Time VALUES (403, '07:00 PM'); 37. INSERT INTO Alumni\_Donation VALUES (1, 401); 38. INSERT INTO Alumni\_Donation VALUES (2, 402); 39. INSERT INTO Alumni\_Donation VALUES (3, 403); 40. INSERT INTO Job VALUES (501, 'Software Engineer', 'Microsoft'); 41. INSERT INTO Job VALUES (502, 'Data Analyst', 'Grameenphone'); 42. INSERT INTO Job VALUES (503, 'AI Researcher', 'Google') 43. INSERT INTO Alumni\_Job VALUES (1, 501); 44. INSERT INTO Alumni\_Job VALUES (2, 502); 45. INSERT INTO Alumni\_Job VALUES (3, 503); 46. INSERT INTO Session\_Date VALUES (601, '09:00 AM', TO\_DATE('2024-08-01','YYYY-MM-DD')); 47. INSERT INTO Session\_Date VALUES (602, '11:00 AM', TO\_DATE('2024-08-01','YYYY-MM-DD')); 48. INSERT INTO Session\_Date VALUES (603, '03:00 PM', TO\_DATE('2024-08-02','YYYY-MM-DD')); 49. INSERT INTO Event\_Session VALUES (201, 601); 50. INSERT INTO Event\_Session VALUES (202, 602); 51. INSERT INTO Event\_Session VALUES (203, 603); 52. INSERT INTO Event\_Batch VALUES (201, 101); 53. INSERT INTO Event\_Batch VALUES (202, 102); 54. INSERT INTO Event\_Batch VALUES (203, 103); |

1. **Query Writing (continuation of Question 3) (Write down the question and the answer. Give full screenshot of the Oracle 10g Homepage that contains the answer and result)**

**-All screenshots MUST include the DATE and TIME feature from the screen of the machine (PC, Laptop etc.) used**

**SQL**

**-2 single-row function**

**-2 group function**

**-2 subquery**

**-2 joining**

***For reference see BasicSQLTutorial and AdvanceSQLTutorial.***

Answer Box 4:

|  |
| --- |
| **Single-row Function:**  **Q1:** **Display all Alumni names in uppercase along with their Email.**  **Answer:**  SELECT UPPER(A\_Name) AS Alumni\_Name, Email  FROM Alumni;    **Q2:** **Display the Alumni name and the year they graduated, but show the first three characters of their name only.**  **Answer:**  SELECT SUBSTR(A\_Name, 1, 3) AS Short\_Name, Graduate\_Year  FROM Alumni;    **Group Function:**  **Q3: Find the total donation amount given by all alumni.**  **Answer:**  SELECT SUM(D\_Amount) AS Total\_Donation  FROM Donation;    **Q4: Count how many Alumni are in each Batch.**  **Answer:**  SELECT Batch\_ID, COUNT(A\_ID) AS Alumni\_Count  FROM Alumni\_Batch  GROUP BY Batch\_ID;    **Subquery:**  **Q5: Find the Alumni names who have donated more than the average donation amount.**  **Answer:**  SELECT A.A\_Name, D.D\_Amount  FROM Alumni A  JOIN Alumni\_Donation AD ON A.A\_ID = AD.A\_ID  JOIN Donation D ON AD.Donation\_ID = D.Donation\_ID  WHERE D.D\_Amount > (SELECT AVG(D\_Amount) FROM Donation);    **Q6:** **Find the Alumni who belong to the same Batch as Alumni with A\_ID = 1.**  **Answer:**  SELECT A.A\_Name  FROM Alumni A  JOIN Alumni\_Batch AB ON A.A\_ID = AB.A\_ID  WHERE AB.Batch\_ID = (  SELECT Batch\_ID  FROM Alumni\_Batch  WHERE A\_ID = 1  );    **Joining:**  **Q7: Display Alumni names with their Event names they attended**.  **Answer:**  SELECT A.A\_Name, E.E\_Name  FROM Alumni A  JOIN Alumni\_Event AE ON A.A\_ID = AE.A\_ID  JOIN Event E ON AE.Event\_ID = E.Event\_ID;    **Q8: Display Alumni names with their Job titles and Company names.**  **Answer:**  SELECT A.A\_Name, J.Job\_Title, J.CompanyName  FROM Alumni A  JOIN Alumni\_Job AJ ON A.A\_ID = AJ.A\_ID  JOIN Job J ON AJ.Job\_ID = J.Job\_ID; |

1. **Query Writing (continuation of Question 4) (Write down the answer only. Give full screenshot of the Oracle 10g Homepage that contains the answer and result)**

**-All screenshots MUST include the DATE and TIME feature from the screen of the machine (PC, Laptop etc.) used**

**PL/SQL**

1. **Convert the SQLs of Question 4 into equivalent PL/SQL code**
2. **For this part, 8 PL/SQL code must be submitted**

Answer Box 5:

|  |
| --- |
| **PL/SQL**  **Single-row Function:**  **Answer Q1:**  BEGIN  FOR rec IN (SELECT UPPER(A\_Name) AS Alumni\_Name, Email  FROM Alumni)  LOOP  DBMS\_OUTPUT.PUT\_LINE('Name: ' || rec.Alumni\_Name || ' | Email: ' || rec.Email);  END LOOP;  END;  /    **Answer Q2:**  BEGIN  FOR rec IN (SELECT SUBSTR(A\_Name, 1, 3) AS Short\_Name, Graduate\_Year  FROM Alumni)  LOOP  DBMS\_OUTPUT.PUT\_LINE('Short Name: ' || rec.Short\_Name || ' | Year: ' || rec.Graduate\_Year);  END LOOP;  END;  /    **Group Function:**  **Answer Q3:**  DECLARE  v\_total NUMBER;  BEGIN  SELECT SUM(D\_Amount)  INTO v\_total  FROM Donation;  DBMS\_OUTPUT.PUT\_LINE('Total Donation: ' || v\_total);  END;  /    **Answer Q4:**  BEGIN  FOR rec IN (SELECT Batch\_ID, COUNT(A\_ID) AS Alumni\_Count  FROM Alumni\_Batch  GROUP BY Batch\_ID)  LOOP  DBMS\_OUTPUT.PUT\_LINE('Batch: ' || rec.Batch\_ID || ' | Alumni Count: ' || rec.Alumni\_Count);  END LOOP;  END;  /    **Subquery:**  **Answer Q5:**  BEGIN  FOR rec IN (  SELECT A.A\_Name, D.D\_Amount  FROM Alumni A  JOIN Alumni\_Donation AD ON A.A\_ID = AD.A\_ID  JOIN Donation D ON AD.Donation\_ID = D.Donation\_ID  WHERE D.D\_Amount > (SELECT AVG(D\_Amount) FROM Donation)  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('Name: ' || rec.A\_Name || ' | Amount: ' || rec.D\_Amount);  END LOOP;  END;  /    **Answer Q6:**  BEGIN  FOR rec IN (  SELECT A.A\_Name  FROM Alumni A  JOIN Alumni\_Batch AB ON A.A\_ID = AB.A\_ID  WHERE AB.Batch\_ID = (SELECT Batch\_ID FROM Alumni\_Batch WHERE A\_ID = 1)  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('Alumni in same Batch: ' || rec.A\_Name);  END LOOP;  END;  /    **Joining:**  **Answer Q7:**  BEGIN  FOR rec IN (  SELECT A.A\_Name, E.E\_Name  FROM Alumni A  JOIN Alumni\_Event AE ON A.A\_ID = AE.A\_ID  JOIN Event E ON AE.Event\_ID = E.Event\_ID  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('Alumni: ' || rec.A\_Name || ' | Event: ' || rec.E\_Name);  END LOOP;  END;  /    **Answer Q8:**  BEGIN  FOR rec IN (  SELECT A.A\_Name, J.Job\_Title, J.CompanyName  FROM Alumni A  JOIN Alumni\_Job AJ ON A.A\_ID = AJ.A\_ID  JOIN Job J ON AJ.Job\_ID = J.Job\_ID  )  LOOP  DBMS\_OUTPUT.PUT\_LINE('Alumni: ' || rec.A\_Name ||  ' | Job: ' || rec.Job\_Title ||  ' | Company: ' || rec.CompanyName);  END LOOP;  END;  / |